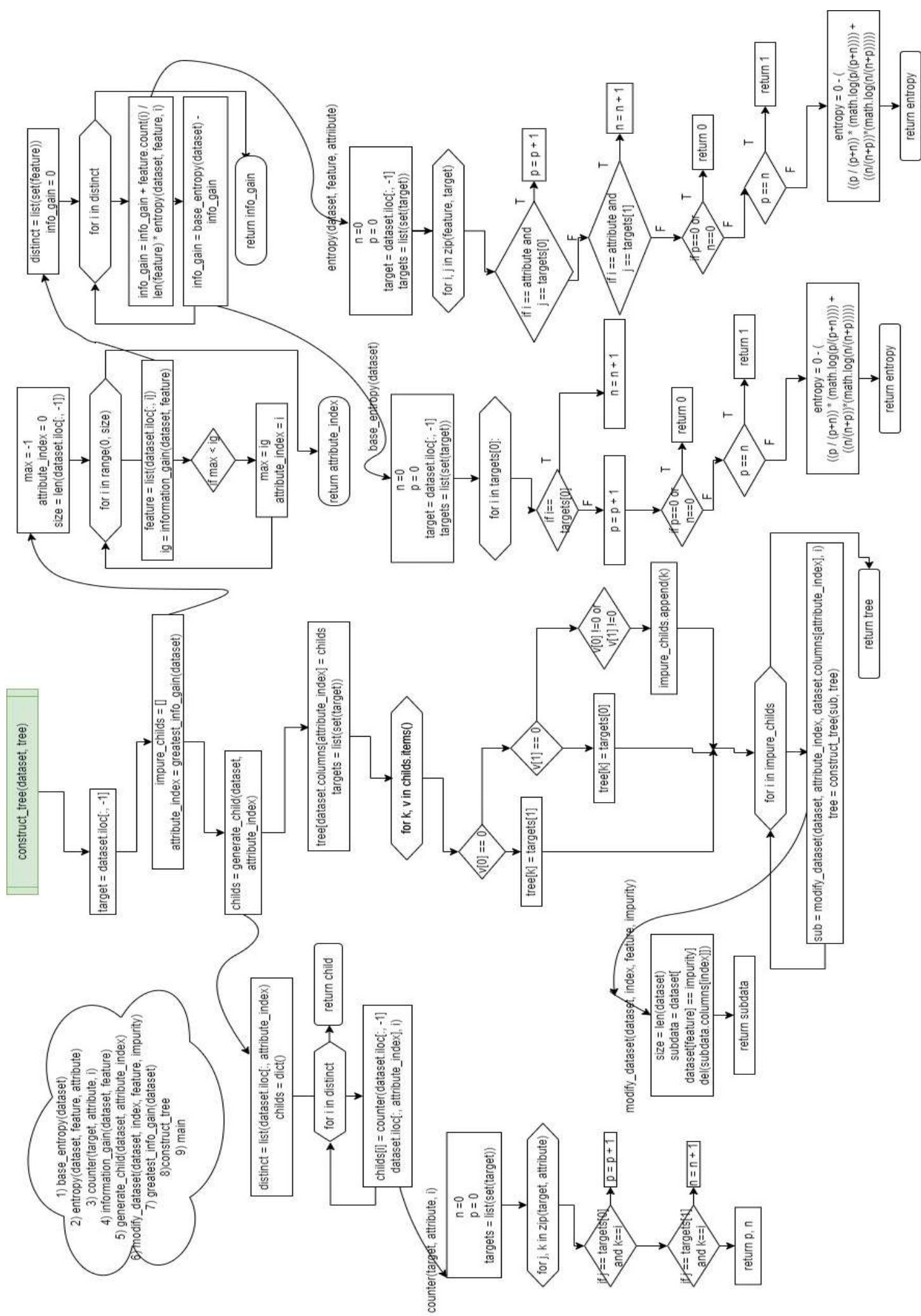
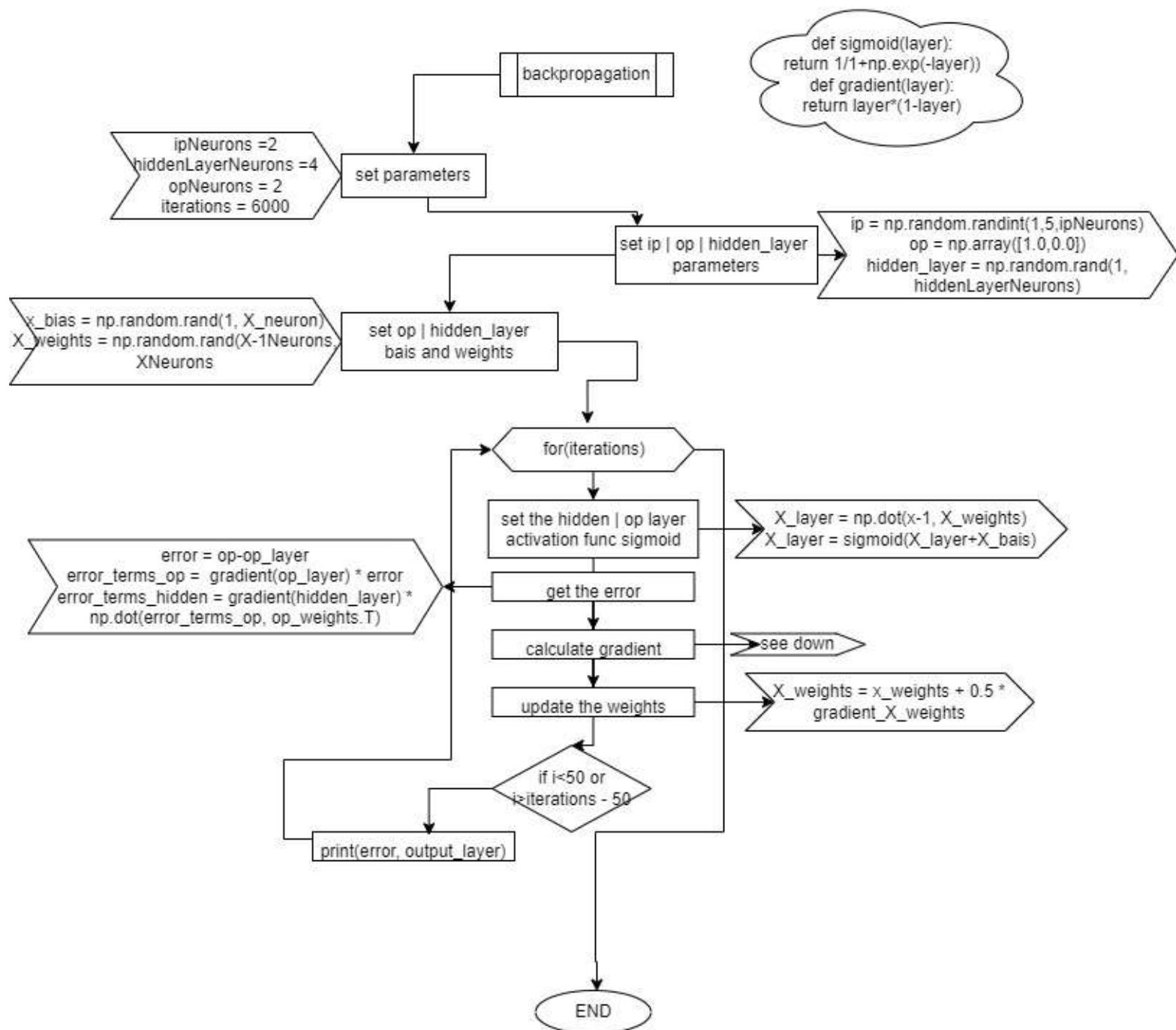


```

gh = []
for i in general:
    for j in i:
        if j != '?':
            gh.append(i)
            break
    print(specific)
    print(gh)
  
```

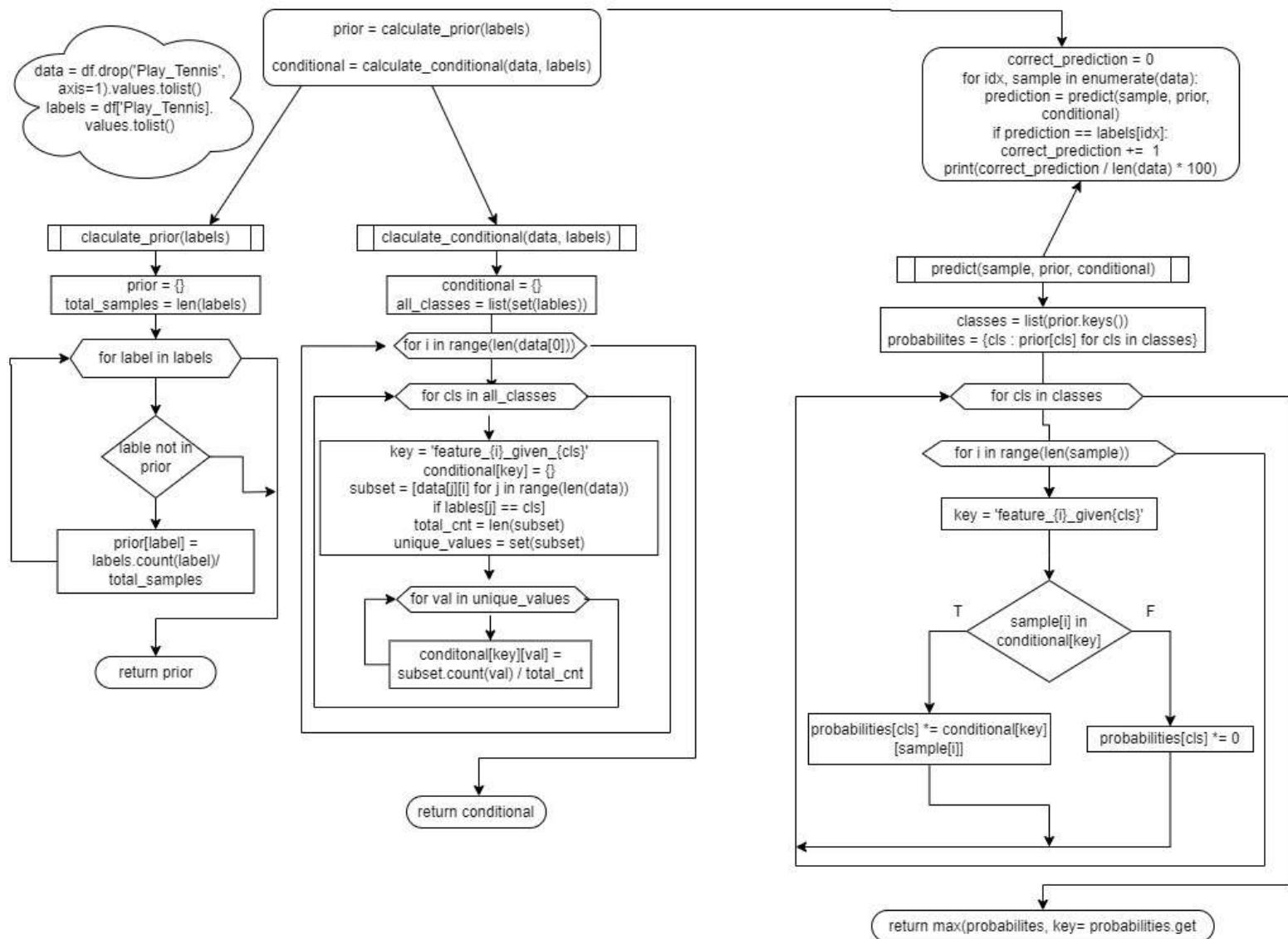


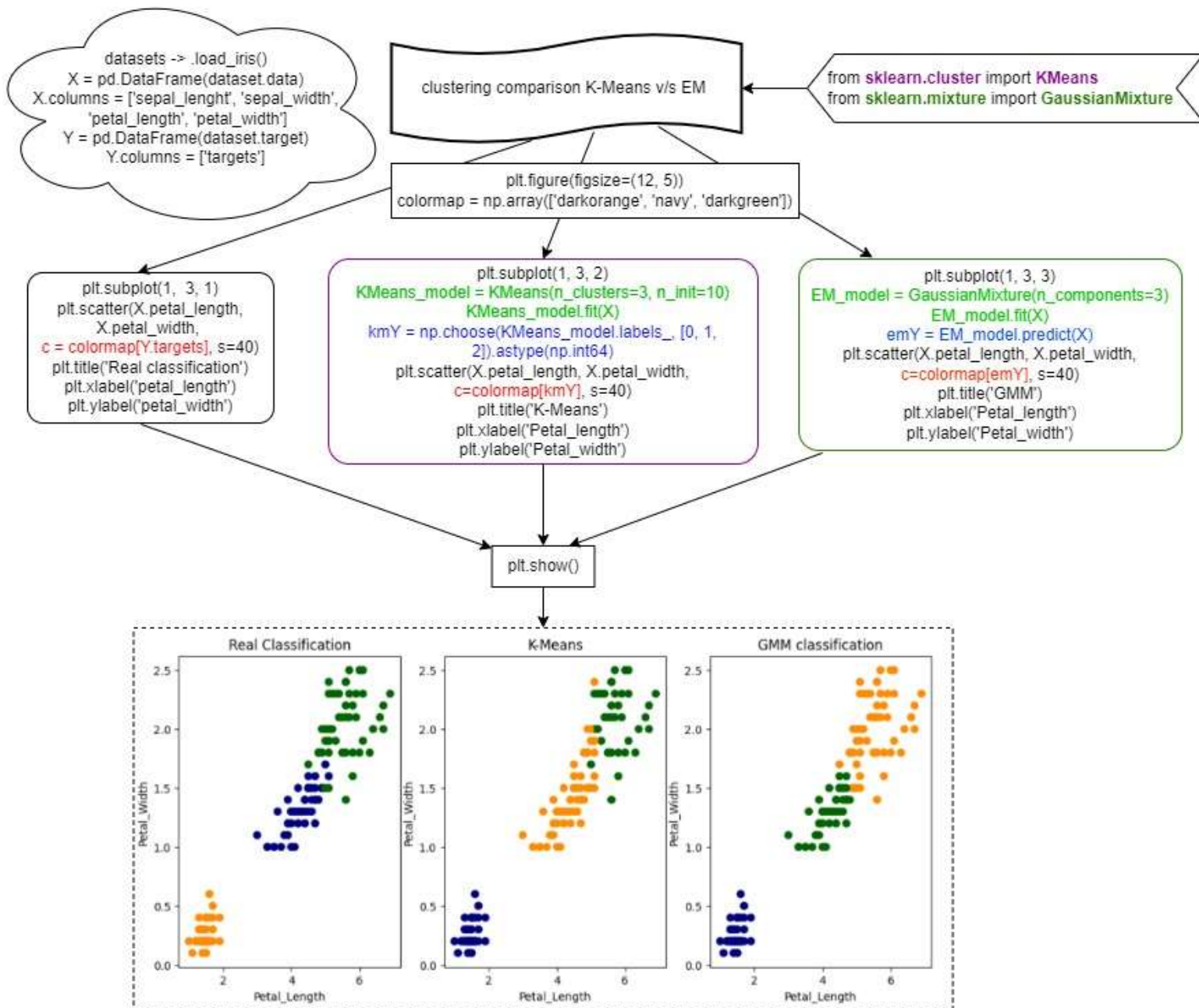


```

gradient_hidden_weights = np.dot(ip.reshape(ipNeurons, 1), error_terms_hidden.reshape(1, hiddenLayerNeurons))
gradient_op_weights = np.dot(hidden_layer.reshape(hiddenLayerNeurons, 1), error_terms_op.reshape(1, opNeurons))

```



0 - Iris-setosa
1 - Iris versicolor
2 - Iris virginica

```
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.model_selection import train_test_split
```

```
X = pd.DataFrame(dataset.data)  
Y = pd.DataFrame(dataset.target)  
x_train, x_test, y_train, y_test = train_test_split(X, Y,  
                                                    test_size = 0.2)
```

```
knn = KNeighborsClassifier(n_neighbors = 5)  
knn.fit(x_train, y_train.values.ravel())
```

```
accuracy_train = knn.score(X_train, Y_train)  
accuracy_test = knn.score(X_test, Y_test)  
print({accuracy_test}, {accuracy_train})
```

```
example = np.array([[4.8, 3.1, 1.2, 0.1]])  
example_prediction = int(knn.predict(example)[0])  
print(getIrisSpecies(example_prediction))
```

```
available_class =  
pd.DataFrame(dataset.target_names)  
print(available_class)
```